

ABSTRACT

Generating unique identifiers (UUIDs) for software objects and other components in a network in which a large number of components may exist simultaneously and/or over a period of time. UUIDs generated by a particular product are divided into two sub-fields.

- 5 One sub-field is stored in non-volatile memory, and incremented infrequently. The other sub-field is stored in volatile memory, that can be incremented frequently. During operation, the product creating the UUIDs generates new UUIDs by incrementing the field stored in volatile memory. When overflow of the volatile memory field occurs, the field stored in non-volatile memory is incremented. A block of non-volatile memory is initialized to all of a first
- 10 logical state, and the bits therein are then sequentially cleared to generate a subsequent unique identifier. The present system provides the equivalent of a counter that can count up to the number of available bits in non-volatile memory plus one, while reducing the number of non-volatile memory erase cycles to one cycle for each time all the bits are cleared.